

Relatório de CT-213: Programação dinâmica

Henrique F. Feitosa

Instituto Tecnológico de Aeronáutica,
São José dos Campos, São Paulo, Brasil

1 Introdução

Nessa prática, buscou-se implementar os algoritmos de programação dinâmica no contexto da solução de um processo decisório de markov. Os algoritmos implementados foram:

1. *policy evaluation*
2. *policy iteration*
3. *value iteration*

O objetivo dessa prática é avaliar as políticas e determinar políticas ótimas para um *grid world*.

Inicialmente, implementou-se o *policy evaluation* seguindo a equação de Bellman e usando uma condição de parada, as quais estão mostrados abaixo.

$$\nu_{\pi}(s) = \sum_{a \in A} \pi(a|s)r(s, a) + \gamma \sum_{a \in A} \sum_{s' \in S} \pi(a|s)p(s'|s, a)\nu_{\pi}(s')$$
$$\max_{s \in S} |\nu_{k+1}(s) - \nu_k(s)| < \varepsilon$$

Após isso, a implementação da iteração de política alterna entre a avaliação de política e o aprimoramento de política. Ademais, a implementação do algoritmo de iteração de valor consiste em iterar diretamente sobre a função valor de acordo com a equação de otimalidade de Bellman:

$$v_*(s) = \max_{a \in A} (r(s, a) + \gamma \sum_{s' \in S} p(s'|s, a)\nu_{\pi}(s'))$$

Finalmente, comparou-se os resultados dos algoritmos implementados para dois casos:

$$p_c = 1.0$$

$$\gamma = 1.0$$

$$p_c = 0.8$$

$$\gamma = 0.98$$

2 Resultados e Discussão

Os resultados obtidos estão nas figuras 1 e 2.

```

Value function:
[ -384.09, -382.73, -381.19, * , -339.93, -339.93]
[ -380.45, -377.91, -374.65, * , -334.92, -334.93]
[ -374.34, -368.82, -359.85, -344.88, -324.92, -324.93]
[ -368.76, -358.18, -346.03, * , -289.95, -309.94]
[ * , -344.12, -315.05, -250.02, -229.99, * ]
[ -359.12, -354.12, * , -200.01, -145.00, 0.00]
Policy:
[ SURDL , SURDL , SURDL , * , SURDL , SURDL ]
[ SURDL , SURDL , SURDL , * , SURDL , SURDL ]
[ SURDL , SURDL , SURDL , SURDL , SURDL , SURDL ]
[ SURDL , SURDL , SURDL , * , SURDL , SURDL ]
[ * , SURDL , SURDL , SURDL , SURDL , * ]
[ SURDL , SURDL , * , SURDL , SURDL , S ]

-----

Value iteration:
Value function:
[ -10.00, -9.00, -8.00, * , -6.00, -7.00]
[ -9.00, -8.00, -7.00, * , -5.00, -6.00]
[ -8.00, -7.00, -6.00, -5.00, -4.00, -5.00]
[ -7.00, -6.00, -5.00, * , -3.00, -4.00]
[ * , -5.00, -4.00, -3.00, -2.00, * ]
[ -7.00, -6.00, * , -2.00, -1.00, 0.00]
Policy:
[ RD , RD , D , * , D , DL ]
[ RD , RD , D , * , D , DL ]
[ RD , RD , RD , R , D , DL ]
[ R , RD , D , * , D , L ]
[ * , R , R , RD , D , * ]
[ R , U , * , R , R , SURD ]

-----

Policy iteration:
Value function:
[ -10.00, -9.00, -8.00, * , -6.00, -7.00]
[ -9.00, -8.00, -7.00, * , -5.00, -6.00]
[ -8.00, -7.00, -6.00, -5.00, -4.00, -5.00]
[ -7.00, -6.00, -5.00, * , -3.00, -4.00]
[ * , -5.00, -4.00, -3.00, -2.00, * ]
[ -7.00, -6.00, * , -2.00, -1.00, 0.00]
Policy:
[ RD , RD , D , * , D , DL ]
[ RD , RD , D , * , D , DL ]
[ RD , RD , RD , R , D , DL ]
[ R , RD , D , * , D , L ]
[ * , R , R , RD , D , * ]
[ R , U , * , R , R , SURD ]

```

Figura 1. Mostra o resultado dos algoritmos para $\gamma = 1$ e $p_e = 1$

```

Value function:
[ -47.19, -47.11, -47.01, * , -45.13, -45.15]
[ -46.97, -46.81, -46.60, * , -44.58, -44.65]
[ -46.58, -46.21, -45.62, -44.79, -43.40, -43.63]
[ -46.20, -45.41, -44.42, * , -39.87, -42.17]
[ * , -44.31, -41.64, -35.28, -32.96, * ]
[ -45.73, -45.28, * , -29.68, -21.88, 0.00]
Policy:
[ SURDL , SURDL , SURDL , * , SURDL , SURDL ]
[ SURDL , SURDL , SURDL , * , SURDL , SURDL ]
[ SURDL , SURDL , SURDL , SURDL , SURDL , SURDL ]
[ SURDL , SURDL , SURDL , * , SURDL , SURDL ]
[ * , SURDL , SURDL , SURDL , SURDL , * ]
[ SURDL , SURDL , * , SURDL , SURDL , S ]
-----
Value iteration:
Value function:
[ -11.65, -10.78, -9.86, * , -7.79, -8.53]
[ -10.72, -9.78, -8.78, * , -6.67, -7.52]
[ -9.72, -8.70, -7.59, -6.61, -5.44, -6.42]
[ -8.70, -7.58, -6.43, * , -4.09, -5.30]
[ * , -6.43, -5.17, -3.87, -2.76, * ]
[ -8.63, -7.58, * , -2.69, -1.40, 0.00]
Policy:
[ D , D , D , * , D , D ]
[ D , D , D , * , D , D ]
[ RD , D , D , R , D , D ]
[ R , RD , D , * , D , L ]
[ * , R , R , D , D , * ]
[ R , U , * , R , R , S ]
-----
Policy iteration:
Value function:
[ -11.65, -10.78, -9.86, * , -7.79, -8.53]
[ -10.72, -9.78, -8.78, * , -6.67, -7.52]
[ -9.72, -8.70, -7.59, -6.61, -5.44, -6.42]
[ -8.70, -7.58, -6.43, * , -4.09, -5.30]
[ * , -6.43, -5.17, -3.87, -2.76, * ]
[ -8.63, -7.58, * , -2.69, -1.40, 0.00]
Policy:
[ D , D , D , * , D , D ]
[ D , D , D , * , D , D ]
[ RD , D , D , R , D , D ]
[ R , RD , D , * , D , L ]
[ * , R , R , D , D , * ]
[ R , U , * , R , R , S ]

```

Figura 2. Mostra o resultado dos algoritmos para $\gamma = 0.98$ e $p_c = 0.8$

Assim, pode-se perceber que a política ótima encontrada foi a mesma para o *policy iteration* e para o *value iteration*, o que era esperado. Além disso, cabe ressaltar que se diminuíssemos o valor do γ e do p_c , é esperado que o *value function* diminua significativamente, uma vez que o retorno considerado se torna menor por causa do γ e o algoritmo ganha uma característica de *expotation*

com a diminuição do p_c , fazendo com que diminua a possibilidade de cair em mínimos locais.